Diversity through Disagreement for Better Transferability

Matteo Pagliardini
EPFL

Martin Jaggi
EPFL

François Fleuret
University of Geneva

Sai Praneeth Karimireddy
EPFL & UC Berkeley

Abstract

Gradient-based learning algorithms have an implicit *simplicity bias* which in effect can limit the diversity of predictors being sampled by the learning procedure. This behavior can hinder the transferability of trained models by (i) favoring the learning of simpler but spurious features — present in the training data but absent from the test data — and (ii) by only leveraging a small subset of predictive features. Such an effect is especially magnified when the test distribution does not exactly match the train distribution—referred to as the Out of Distribution (OOD) generalization problem. However, given only the training data, it is not always possible to apriori assess if a given feature is spurious or transferable. Instead, we advocate for learning an ensemble of models which capture a diverse set of predictive features. Towards this, we propose a new algorithm D-BAT (Diversity-By-disAgreement Training), which enforces agreement among the models on the training data, but disagreement on the OOD data. We show how D-BAT naturally emerges from the notion of generalized discrepancy, as well as demonstrate in multiple experiments how the proposed method can mitigate shortcut-learning, enhance uncertainty and OOD detection, as well as improve transferability. A more detailed version of our work can be found via the following link: https://arxiv.org/abs/2202.04414.

1 Introduction

While gradient-based learning algorithms such as Stochastic Gradient Descent (SGD), are nowadays ubiquitous in the training of Deep Neural Networks (DNNs), it is well known that the resulting models are (i) brittle when exposed to small distribution shifts (Beery et al., 2018; Sun et al., 2016; Amodei et al., 2016), (ii) can easily be fooled by small adversarial perturbations (Szegedy et al., 2014), (iii) tend to pick up spurious correlations (McCoy et al., 2019; Oakden-Rayner et al., 2020; Geirhos et al., 2020) — present in the training data but absent from the downstream task — , as well as (iv) fail to provide adequate uncertainty estimates (Kim et al., 2016; van Amersfoort et al., 2020; Liu et al., 2021b). Recently those learning algorithms have been investigated for their implicit bias toward simplicity — known as Simplicity Bias (SB), seen as one of the reasons behind their superior generalization properties (Arpit et al., 2017; Dziugaite & Roy, 2017). While for deep neural networks, simpler decision boundaries are often seen as less likely to overfit, Shah et al. (2020); Pezeshki et al. (2021) demonstrated that the SB can still cause the aforementioned issues. In particular, they show how the SB can be *extreme*, compelling predictors to rely only on the simplest feature available, despite the presence of equally or even more predictive complex features.

Correspondence to matteo.pagliardini@epfl.ch and sp.karimireddy@berkeley.edu

Workshop on Distribution Shifts, 36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Its effect is greatly increased when we consider the more realistic out of distribution (OOD) setting (Ben-Tal et al., 2009), in which the source and target distributions are different, known to be a challenging problem (Sagawa et al., 2020; Krueger et al., 2021). The difference between the two domains can be categorized into either a distribution shift — e.g. a lack of samples in certain parts of the data manifold due to limitations of the data collection pipeline —, or as simply having completely different distributions. In the first case, the SB in its extreme form would increase the chances of learning to rely on spurious features — shortcuts not generalizing to the target distribution. Classic manifestations of this in vision applications are when models learn to rely mostly on textures or backgrounds instead of more complex and likely more generalizable semantic features such as using shapes (Beery et al., 2018; Ilyas et al., 2019; Geirhos et al., 2020). In the second instance, by relying only on the simplest feature, and being invariant to more complex ones, the SB would cause confident predictions (low uncertainty) on completely OOD samples. This even if complex features are contradicting simpler ones. Which brings us to our goal of deriving a method which can (i) learn more transferable features, better suited to generalize despite distribution shifts, and (ii) provides accurate uncertainty estimates also for OOD samples.

We aim to achieve those two objectives through learning an ensemble of diverse predictors $(h_1, \ldots, h_K)$, with $h : \mathcal{X} \to \mathcal{Y}$, and K being the ensemble size. Suppose that our training data is drawn from the distribution $\mathcal{D}$, and $\mathcal{D}_{\text{ood}}$ is the distribution of OOD data on which we will be tested. Importantly, $\mathcal{D}$ and $\mathcal{D}_{\text{ood}}$ may have non-overlapping support, and $\mathcal{D}_{\text{ood}}$ is not known during training. Our proposed method, D-BAT (Diversity-By-disAgreement Training), relies on the following idea:

> *Diverse hypotheses should agree on the source distribution $\mathcal{D}$ while disagreeing on the OOD distribution $\mathcal{D}_{ood}$.*

Intuitively, a set of hypotheses should agree on what is known i.e. on $\mathcal{D}$, while formulating different interpretations of what is *not* known, i.e. on $\mathcal{D}_{\text{ood}}$. Even if each *individual predictor* might be wrongly confident on OOD samples, while predicting different outcomes — the resulting uncertainty of the *ensemble* on those samples will be increased. Disagreement on $\mathcal{D}_{\text{ood}}$ can itself be enough to promote learning diverse representations of instances of $\mathcal{D}$. In the context of object detection, if one model h_1 is relying on textures only, this model will generate predictions on $\mathcal{D}_{\text{ood}}$ based on textures, when enforcing disagreement on $\mathcal{D}_{\text{ood}}$, a second model h_2 would be discouraged to use textures in order to disagree with h_1 — and consequently look for a different hypothesis to classify instances of $\mathcal{D}$ e.g. using shapes. Once trained, the ensemble can either be used by forming a weighted average of the probability distribution from each hypothesis, or by tuning the weights on a downstream task.

Contributions. Our results can be summarized as:

- We introduce the D-BAT method to train ensembles of diverse predictors.
- We show how the induced diversity can help to (i) tackle shortcut learning, and (ii) improve uncertainty estimation and transferability.

2 Related Work

Simplicity bias. Recently Teney et al. (2021) proposed to evade the simplicity bias by adding gradient orthogonality constrains, not at the output level, but at an intermediary hidden representation obtained after a shared and fixed encoder. See App. E.1 for a detailed comparison.

OOD generalization. Generalizing to distributions not seen during training is accomplished by two approaches: robust training, and invariant learning. In the former, the test distribution is assumed to be within a set of known plausible distributions (say $\mathcal{U}$). Then, robust training minimizes the loss over the worst possible distribution in $\mathcal{U}$ (Ben-Tal et al., 2009). Numerous approaches exist to defining the set $\mathcal{U}$ - see survey by (Rahimian & Mehrotra, 2019). Most recently, Sagawa et al. (2020) model the set of plausible domains as the convex hull over predefined subgroups of datapoints and Krueger et al. (2021) extend this by taking affine combinations beyond the convex hull. A different line of work defines a set of environments and asks that our outputs be 'invariant' among the different environments (Bengio et al., 2013; Arjovsky et al., 2019; Koyama & Yamaguchi, 2020). Prior work in OOD generalization are often considering datasets where the spurious feature is *not* fully predictive in the training distribution (Zhang et al., 2021; Saito et al., 2017; Nam et al., 2020; Liu et al., 2021a), and fail in our challenging settings of § 4.1, see App. E for a more detailed comparison.

Uncertainty estimation. DNNs are notoriously unable to provide reliable confidence estimates, which is impeding the progress of the field in safety critical domains (Begoli et al., 2019), as well as hurting models interpretability (Kim et al., 2016). To improve the confidence estimates of DNNs, Gal & Ghahramani (2016) propose to use dropout at inference time, a method referred to as MC-Dropout. Other popular methods used for uncertainty estimation are Bayesian Neural Networks (BNNs) (Hernández-Lobato & Adams, 2015) and Gaussian Processes (Rasmussen & Williams, 2005). All those methods but gaussian processes, were recently shown to fail to adequately provide high uncertainty estimates on OOD samples *away* from the boundary decision (van Amersfoort et al., 2020; Liu et al., 2021b). D-BAT can help to associate high uncertainty to those samples.

3 Diversity through Disagreement

3.1 Motivating D-BAT

Setup. Let us formally define the OOD problem. $\mathcal{X}$ is the input space, $\mathcal{Y}$ the output space, we define a domain as a pair of a distribution over $\mathcal{X}$ and a labeling function $h : \mathcal{X} \to \mathcal{Y}$. Given any distribution $\mathcal{D}$ over $\mathcal{X}$, given two labeling functions h_1 and h_2, given a loss function $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$, we define the expected loss as the expectation: $\mathcal{L}_\mathcal{D}(h_1, h_2) = \mathbb{E}_{x\sim\mathcal{D}}[L(h_1(x), h_2(x))]$.

Now, suppose that the training data is drawn from a source distribution $(\mathcal{D}_s, h_\text{s})$, but we will be tested on a different distribution $(\mathcal{D}_\text{ood}, h_\text{ood})$. While the labelling function h_ood is unknown, we assume that we have access to unlabelled samples from $\mathcal{D}_\text{ood}$. Finally, let $\mathcal{H}$ be the set of all labelling functions i.e. the set of all possible prediction models. And further define $\mathcal{H}_s^\star$ and $\mathcal{H}_\text{ood}^\star$ to be the optimal labelling functions on the train and the OOD domains:

$$\mathcal{H}_s^\star := \arg\min_{h\in\mathcal{H}} \mathcal{L}_{\mathcal{D}_s}(h, h_\text{s}),\, \mathcal{H}_\text{ood}^\star := \arg\min_{h\in\mathcal{H}} \mathcal{L}_{\mathcal{D}_\text{ood}}(h, h_\text{ood}).$$

We assume that there exists an ideal transferable function $h^\star \in \mathcal{H}_s^\star \cap \mathcal{H}_\text{ood}^\star$. This assumption captures the reality that the training task and the OOD testing task are closely related to each other. Otherwise, we would not expect any OOD generalization.

Beyond standard training. Just using the training data, standard training would train a model $h_\text{ERM} \in \mathcal{H}_s^\star$. However, as we discussed in the introduction, if we use gradient descent to find the ERM solution, then h_ERM will likely be the simplest model i.e. it will likely pick up spurious correlations in $\mathcal{D}_s$ which are not present in $\mathcal{D}_\text{ood}$. Thus, the error on OOD data might be very high. More formally, not knowing $\mathcal{H}_\text{ood}^\star$, we do not know how far we are from $h^\star$:

$$\mathcal{L}_{\mathcal{D}_\text{ood}}(h_\text{ERM}, h^\star) \leq \max_{h\in\mathcal{H}_s^\star} \mathcal{L}_{\mathcal{D}_\text{ood}}(h, h^\star)\,.$$

We can however upper-bound the quantity on the left in the following way:

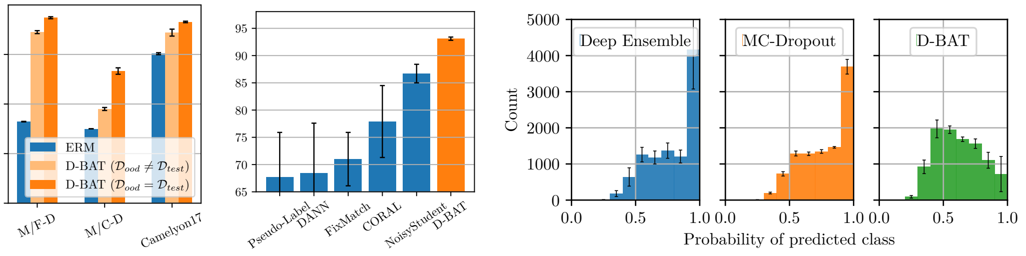
$$\mathcal{L}_{\mathcal{D}_\text{ood}}(h_1, h^\star) \leq \max_{h_2\in\mathcal{H}_s^\star} \mathcal{L}_{\mathcal{D}_\text{ood}}(h_1, h_2)$$

In the above we used the fact that $\mathcal{H}_s^\star \cap \mathcal{H}_\text{ood}^\star$ is non-empty. Recall that $\mathcal{H}_s^\star = \arg\min_{h\in\mathcal{H}} \mathcal{L}_{\mathcal{D}_s}(h, h_\text{s})$. So this means we can use the minimization of the quantity on the right as a proxy to minimize $\mathcal{L}_{\mathcal{D}_\text{ood}}(h_1, h^\star)$. This means we want to pick h_2 to minimize our training data (i.e. belong to $\mathcal{H}_s^\star$), but otherwise maximally disagree with h_1 on the OOD data. The latter is closely related to the concept of discrepancy in domain-adaption (Mansour et al., 2009; Cortes et al., 2019). However, the main difference between the definitions is that we restrict the maximum to the set of $\mathcal{H}_s^\star$, whereas the standard notions use an unrestricted maximum. Thus, our version is tighter when the train and OOD tasks are closely related.

3.2 Algorithm description

Binary classification formulation. Concretely given a binary classification task, with $\mathcal{Y} = \{0, 1\}$, we train two models sequentially. The training of the first model h_1 is done in a classical way, minimizing its empirical classification loss $\mathcal{L}(h_1(\boldsymbol{x}), y)$ over samples $(\boldsymbol{x}, y)$ from $\hat{\mathcal{D}}$. Once h_1 trained, we train the second model h_2 adding a term $\mathcal{A}_{\tilde{\boldsymbol{x}}}(h_1, h_2)$ representing the agreement on samples $\tilde{\boldsymbol{x}}$ of $\hat{\mathcal{D}}_\text{ood}$, with some weight $\alpha \geq 0$:

$$h_2^\star \in \underset{h_2\in\mathcal{H}}{\text{argmin}}\, \frac{1}{N}\Big(\sum_{(\boldsymbol{x},y)\in\hat{\mathcal{D}}} \mathcal{L}(h_2(\boldsymbol{x}), y) + \alpha \sum_{\tilde{\boldsymbol{x}}\in\hat{\mathcal{D}}_\text{ood}} \mathcal{A}_{\tilde{\boldsymbol{x}}}(h_1, h_2)\Big)$$



(a) D-BAT vs. ERM (b) Camelyon17 (c) CIFAR-10

Figure 1: **(a)**: Comparison of test accuracies between D-BAT and ERM. We consider the two cases in which unlabeled test data is available and can be used as $\mathcal{D}_{\text{ood}}$, and the case where $\mathcal{D}_{\text{ood}}$ is different from the test distribution. In both cases D-BAT improves the test accuracy. **(b)**: Comparison of test-accuracies of D-BAT with several other methods on the Camelyon17 dataset, baselines are taken from Sagawa et al. (2022). **(c)**: Histogram of predicted probabilities on OOD data. For both deep ensembles and MC-Dropout a large amount of predicted probabilities are larger than 0.9, which indicate those methods are overly confident on OOD data. In contrast, most of the predicted probabilities of D-BAT ensembles are smaller than 0.7. The average ensemble accuracies for all those methods are 92% for deep ensembles, 91.2% for D-BAT ensembles, and 90.4% for MC-Dropout.

Given $p_{h,\boldsymbol{x}}^{(y)}$ the probability of class y predicted by h given $\boldsymbol{x}$, the agreement $\mathcal{A}_{\tilde{\boldsymbol{x}}}(h_1, h_2)$ is defined as:

$$\mathcal{A}_{\tilde{\boldsymbol{x}}}(h_1, h_2) = -\log\left(p_{h_1,\tilde{\boldsymbol{x}}}^{(0)} \cdot p_{h_2,\tilde{\boldsymbol{x}}}^{(1)} + p_{h_1,\tilde{\boldsymbol{x}}}^{(1)} \cdot p_{h_2,\tilde{\boldsymbol{x}}}^{(0)}\right) \tag{AG}$$

The binary classification formulation of D-BAT is straightforward, a multi-class version is also straightforward and detailed in App. B.

4 Experiments

4.1 OOD generalization and Avoiding shortcuts

Experimental setup. We consider two types of datasets: (i) artificial datasets for which we know the spurious correlations in the style of Teney et al. (2021), and (ii) natural datasets for which spurious correlations are unknown. We build the M/F-D artifical dataset, resp. M/C-D, by concatenating MNIST (Lecun & Cortes, 1998) images of 0s and 1s with images of Fashion-MNIST (Xiao et al., 2017) corresponding to coats and dresses, resp. images of CIFAR10 (Krizhevsky, 2009) corresponding to trucks and cars. For those datasets the spuriousness of the MNIST feature is total, i.e. it predicts perfectly the label for samples of the training distribution but not for samples of the test distribution. In the test distributions the MNIST mnist images of 0s and 1s are picked randomly. Due to the simplicity bias, an ERM model would only learn to use the MNIST feature and do no better than random on the test distribution. Samples from those datasets can be seen in App. D. As natural dataset, we use the Camelyon17 (Bandi et al., 2018) dataset, a cancer detection dataset where images are gathered from different hospitals, each hospital being uniquely associated to the training, validation or test splits. For both settings we use either unlabeled test data as $\mathcal{D}_{\text{ood}}$ or data from a different domain: unlabeled validation data in the case of Camelyon17, and concatenation of MNIST 0s and 1s with random classes from Fashion-MNIST and CIFAR10 not present in the training data. Results of those experiments can be seen in Fig. 1.a and Fig. 1.b.

4.2 Better Uncertainty & OOD Detection

CIFAR-10 setup. We train ensembles of 4 models and benchmark three different methods in their ability to identify what they do not know. For this we look at the histograms of the probability of their predicted classes on OOD samples. As training set we use the CIFAR-10 classes $\{0, 1, 2, 3, 4\}$. We use the CIFAR-100 (Krizhevsky, 2009) test set as OOD samples to compute the histograms. For D-BAT we use the remaining CIFAR-10 classes, $\{5, 6, 7, 8, 9\}$, as $\mathcal{D}_{\text{ood}}$, and set α to 0.2. Histograms are averaged over 5 seeds. The three methods considered are simple deep-ensembles (Lakshminarayanan et al., 2017), MC-Dropout models (Gal & Ghahramani, 2016), and D-BAT ensembles. We use a modified ResNet-18 (He et al., 2016) with added dropout to accommodate MC-Dropout, we use a dropout probability of 0.2 for the three methods. For MC-Dropout, we sample 20 distributions. See Fig. 1 for results.

5 Discussion

In this work we introduced D-BAT, a novel training method to promote diversity in ensembles of predictors. By encouraging disagreement on OOD data, while agreeing on the training data, we effectively (i) give incentives to our predictors to rely on diverse features, (ii) which enhance the transferability of the ensemble and (iii) improve uncertainty estimation and OOD detection.

References

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul F. Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *CoRR*, abs/1606.06565, 2016.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Devansh Arpit, Stanisław Jastrzundefinedbski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In *ICML*, pp. 233–242. JMLR, 2017.

Peter Bandi, Oscar Geessink, Quirine Manson, Marcory Van Dijk, Maschenka Balkenhol, Meyke Hermsen, Babak Ehteshami Bejnordi, Byungjae Lee, Kyunghyun Paeng, Aoxiao Zhong, et al. From detection of individual metastases to classification of lymph node status at the patient level: the camelyon17 challenge. *IEEE Transactions on Medical Imaging*, 2018.

Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *ECCV (16)*, volume 11220 of *Lecture Notes in Computer Science*, pp. 472–489. Springer, 2018.

Edmon Begoli, Tanmoy Bhattacharya, and Dimitri Kusnezov. The need for uncertainty quantification in machine-assisted medical decision making. *Nat. Mach. Intell.*, 1(1):20–23, 2019.

Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust Optimization*, volume 28 of *Princeton Series in Applied Mathematics*. Princeton University Press, 2009.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

Corinna Cortes, Mehryar Mohri, and Andrés Munoz Medina. Adaptation based on generalized discrepancy. *The Journal of Machine Learning Research*, 20(1):1–30, 2019.

Gintare Karolina Dziugaite and Daniel M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *Proceedings of the 33rd Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, volume 48 of *JMLR Workshop and Conference Proceedings*, pp. 1050–1059. JMLR.org, 2016.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard S. Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nat. Mach. Intell.*, 2(11):665–673, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

José Miguel Hernández-Lobato and Ryan P. Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *ICML*, volume 37 of *JMLR Workshop and Conference Proceedings*, pp. 1861–1869. JMLR.org, 2015.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *arXiv preprint arXiv:1905.02175*, 2019.

Been Kim, Oluwasanmi Koyejo, and Rajiv Khanna. Examples are not enough, learn to criticize! criticism for interpretability. In *NIPS*, pp. 2280–2288, 2016.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton A. Earnshaw, Imran S. Haque, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning (ICML)*, 2021.

Masanori Koyama and Shoichiro Yamaguchi. Out-of-distribution generalization with maximal invariant predictor. *arXiv preprint arXiv:2008.01883*, 2020.

Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Master's thesis, 2009.

David Krueger, Ethan Caballero, Jörn-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Rémi Le Priol, and Aaron C. Courville. Out-of-distribution generalization via risk extrapolation (rex). In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5815–5826. PMLR, 2021.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 6405–6416, Red Hook, NY, USA, 2017. Curran Associates Inc.

Yann Lecun and Corinna Cortes. The MNIST database of handwritten digits. 1998. URL http://yann.lecun.com/exdb/mnist/.

Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pp. 2278–2324, 1998.

Yoonho Lee, Huaxiu Yao, and Chelsea Finn. Diversify and disambiguate: Learning from underspecified data. *CoRR*, abs/2202.03418, 2022.

Evan Zheran Liu, Behzad Haghgoo, Annie S. Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6781–6792. PMLR, 2021a.

Yehao Liu, Matteo Pagliardini, Tatjana Chavdarova, and Sebastian U. Stich. The peril of popular deep learning uncertainty estimation methods. 2021b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR (Poster)*. OpenReview.net, 2019.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. *arXiv preprint arXiv:0902.3430*, 2009.

Tom McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *ACL (1)*, pp. 3428–3448. Association for Computational Linguistics, 2019.

Jun Hyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: Training debiased classifier from biased classifier. *CoRR*, abs/2007.02561, 2020.

Luke Oakden-Rayner, Jared Dunnmon, Gustavo Carneiro, and Christopher Ré. Hidden stratification causes clinically meaningful failures in machine learning for medical imaging. In *CHIL*, pp. 151–159. ACM, 2020.

Mohammad Pezeshki, Sékou-Oumar Kaba, Yoshua Bengio, Aaron C. Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. In *NeurIPS*, pp. 1256–1272, 2021.

Hamed Rahimian and Sanjay Mehrotra. Distributionally robust optimization: A review. *arXiv preprint arXiv:1908.05659*, 2019.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In *ICLR*. OpenReview.net, 2020.

Shiori Sagawa, Pang Wei Koh, Tony Lee, Irena Gao, Sang Michael Xie, Kendrick Shen, Ananya Kumar, Weihua Hu, Michihiro Yasunaga, Henrik Marklund, Sara Beery, Etienne David, Ian Stavness, Wei Guo, Jure Leskovec, Kate Saenko, Tatsunori Hashimoto, Sergey Levine, Chelsea Finn, and Percy Liang. Extending the WILDS benchmark for unsupervised adaptation. In *ICLR*. OpenReview.net, 2022.

Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Asymmetric tri-training for unsupervised domain adaptation. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2988–2997. PMLR, 2017.

Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, pp. 3723–3732. Computer Vision Foundation / IEEE Computer Society, 2018.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *AAAI*, pp. 2058–2065. AAAI Press, 2016.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR (Poster)*, 2014.

Damien Teney, Ehsan Abbasnejad, Simon Lucey, and Anton van den Hengel. Evading the simplicity bias: Training a diverse set of models discovers solutions with superior OOD generalization. *CoRR*, abs/2105.05612, 2021.

Joost van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9690–9700. PMLR, 2020.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv:1708.07747*, 2017.

Dinghuai Zhang, Kartik Ahuja, Yilun Xu, Yisen Wang, and Aaron C. Courville. Can subnetwork structure be the key to out-of-distribution generalization? In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12356–12367. PMLR, 2021.

A Source code

Link to the source code to reproduce our experiments: https://github.com/mpagli/Agree-to-Disagree

B Algorithms

The D-BAT training algorithm can be applied to both binary and multi-class classification problems. For our experiments on binary classification — as for the Camelyon17, M/F-D, M/C-D — we used Alg. 1. This algorithm assumes a first model h_1 has already been trained with e.g. empirical risk minimization, and trains a second model following the algorithm described in § 3.2. For our multi-class experiments — as for the CIFAR-10 uncertainty experiments (see § 4.2), we used Alg. 2. This algorithm is training a full ensemble of size M using D-BAT as described in § 3.2.

Algorithm 1 D-BAT for binary classification

```
  **Input:** train data $\mathcal{D}$, OOD data $\mathcal{D}_{\text{ood}}$, stopping time $T$, D-BAT coefficient $\alpha$, learning rate $\eta$, pre-trained model $h_1$, randomly initialized model $h_2$ with weights $\boldsymbol{\omega}_0$, and its loss $\mathcal{L}$.
  **for** $t \in 0, \dots, T-1$ **do**
    **Sample** $(\boldsymbol{x}, y) \sim \mathcal{D}$
    **Sample** $\tilde{\boldsymbol{x}} \sim \mathcal{D}_{\text{ood}}$
    $\boldsymbol{\omega}_{t+1} = \boldsymbol{\omega}_t - \eta \nabla_{\boldsymbol{\omega}} \big( \mathcal{L}(h_2, \boldsymbol{x}, y) + \alpha \mathcal{A}(h_1, h_2, \tilde{\boldsymbol{x}}) \big)$
  **end for**
```

Algorithm 2 D-BAT for multi-class classification

```
  **Input:** ensemble size $M$, train data $\mathcal{D}$, OOD data $\mathcal{D}_{\text{ood}}$, stopping time $T$, D-BAT coefficient $\alpha$, learning rate $\eta$, randomly initialized models $(h_0, \dots, h_{M-1})$ with resp. weights $(\boldsymbol{\omega}_0^{(0)}, \dots, \boldsymbol{\omega}_0^{(M-1)})$, and a classification loss $\mathcal{L}$.
  **for** $m \in 0, \dots, M-1$ **do**
    **for** $t \in 0, \dots, T-1$ **do**
      **Sample** $(\boldsymbol{x}, y) \sim \mathcal{D}$
      **Sample** $\tilde{\boldsymbol{x}} \sim \mathcal{D}_{\text{ood}}$
      $\mathcal{A} \leftarrow 0$
      $\tilde{y} \leftarrow \text{argmax} h_m(\tilde{\boldsymbol{x}})$
      **for** $i \in 0, \dots, m-1$ **do**
        $\mathcal{A} = \mathcal{A} - \frac{1}{m-1} \log \left( p^{(\tilde{y})}_{h_i, \tilde{\boldsymbol{x}}} \cdot p^{(\neg \tilde{y})}_{h_m, \tilde{\boldsymbol{x}}} + p^{(\neg \tilde{y})}_{h_i, \tilde{\boldsymbol{x}}} \cdot p^{(\tilde{y})}_{h_m, \tilde{\boldsymbol{x}}} \right)$
      **end for**
      $\boldsymbol{\omega}^{(m)}_{t+1} = \boldsymbol{\omega}^{(m)}_t - \eta \nabla_{\boldsymbol{\omega}^{(m)}} \big( \mathcal{L}(h_m, \boldsymbol{x}, y) + \alpha \mathcal{A} \big)$
    **end for**
  **end for**
```

Sequential vs. simultaneous training. Nothing prevents the use of the D-BAT objective while training all the predictors of the ensemble simultaneously. While we had some successes in doing so, we advocate against it as this can discard the ERM solution. We found that the training dynamics of simultaneous training have a tendency to generate more complex solutions than sequential training. In our experiments on the 2D toy setting, sequential training gives two models which are both simple and diverse, whereas simultaneous training generates two relatively simple predictors but of higher complexity, especially it would deprive us from the simplest solution. In general as we do not know the spuriousness of the features, the simplest predictor is still of importance.

C Omitted details on experiments

C.1 Implementation details for the M/F-D and M/C-D experiments

In the experiments on M/F-D and M/C-D, we used different versions of LeNet (Lecun et al., 1998):

- For the MF-Dominoes datasets, we increase the input dimension of the first fully-connected layer to 960.

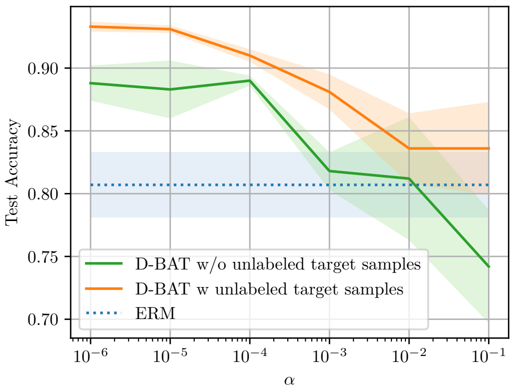


Figure 2: Test accuracy given α. We compare the best ERM model with the second model trained using D-BAT, for varying α hyperparameters.

- For the MC-Dominoes dataset, we use 3 input channels, increase the number of output channels of the first convolution to 32, and of the second one to 56. We modify the fully-connected layers to be $2016 \to 512 \to 256 \to c$ with c the number of classes.

We use the AdamW optimizer (Loshchilov & Hutter, 2019) for all our experiments. For all the datasets in this section, we only train ensembles of 2 models, which we denote $\mathcal{M}_1$ and $\mathcal{M}_2$. We select the model with best validation accuracy. When building the OOD datasets, we make sure the images used are not shared with the images used to build the training, test and validation sets. Our results are obtained by averaging over 5 seeds. For further details on the implementation, we invite the reader to check the source code, see § A.

C.2 Implementation details for the Camelyon17 experiments

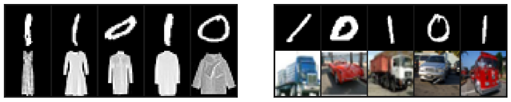
The CameLyon17 cancer detection dataset (Bandi et al., 2018) is taken from the WILDS collection (Koh et al., 2021). The dataset consists of a training, validation, and test sets of images coming from different hospitals, each hospital being uniquely associated to a given split. The goal is to generalize to hospitals not necessarily present in the training set.

We use a ResNet-50 (He et al., 2016) as model. We train for 60 epochs with a fixed learning rate of 0.001 with and SGD as optimizer. We an l_2 penalty term of 0.0001 and a momentum term $\beta = 0.9$. For D-BAT, we tune $\alpha \in \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$ and found $\alpha = 10^{-6}$ to be best. For each set of hyperparameters, we train a deep-ensemble and a D-BAT ensemble of size 2, and select the parameters associated with the highest averaged validation accuracy over the two predictors of the ensemble. Our results are obtained by averaging over 3 seeds.

In Fig. 2, we plot the evolution of the test accuracy as a function of α for both setups discussed in § 4.1. In the first "ideal" setup we have access to unlabeled target data to use as $\hat{\mathcal{D}}_{\text{ood}}$. In the second setup we do not, instead we use samples from different hospitals. In the case of the Camelyon dataset, we use the available unlabeled validation data. Despite this data belonging to a different domain, we still get a significant improvement in test accuracy.

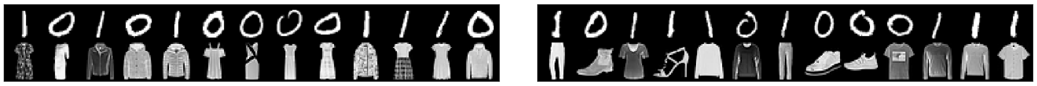
D Training and OOD distribution samples M/F-D and M/C-D

In Fig. 3, we show some samples from some of the training distribution used in § 4.1. We also introduce the MM-Dominoes dataset, similar in spirit to the other dominoes dataset but concatenating MNIST digits of `0`s and `1`s with MNSIT digits `7` and `9`. In Figs. 4,5, we show samples for the OOD distributions used in § 4.1.



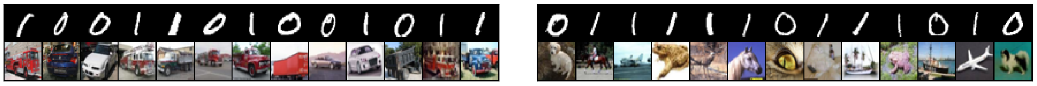
(a) MF-Dominoes (b) MC-Dominoes

Figure 3: Samples from the training data distribution $\mathcal{D}$ for M/F-D, and M/C-D. Those datasets are used to evaluate D-BAT's aptitude to evade the simplicity bias. For all the Dominoes datasets, the simple feature is the top row, while the complex feature is the bottom one. One could indeed separate `0`s from `1`s by simply looking at the value of the middle pixels (if low value then `0` else `1`).



(a) $\mathcal{D}_{\text{ood}}^{(1)}$ (b) $\mathcal{D}_{\text{ood}}^{(2)}$

Figure 4: OOD distributions used for the MF-Dominoes experiments. $\mathcal{D}_{\text{ood}}^{(1)}$ corresponds to our experiments when we have access to unlabeled target data. $\mathcal{D}_{\text{ood}}^{(2)}$ is very different from the target distribution as the second row is made only of images from categories not present in the training and test distributions.



(a) $\mathcal{D}_{\text{ood}}^{(1)}$ (b) $\mathcal{D}_{\text{ood}}^{(2)}$

Figure 5: OOD distributions used for the MC-Dominoes experiments. $\mathcal{D}_{\text{ood}}^{(1)}$ corresponds to our experiments when we have access to unlabeled target data. $\mathcal{D}_{\text{ood}}^{(2)}$ is very different from the target distribution as the second row is made only of images from categories not present in the training and test distributions.

E Additional discussions and experiments

When two features are equally predictive but have different complexities, the more complex feature will be discarded due to the extreme simplicity bias. This happens despite the uncertainty over the potential spuriousness of the simpler feature. For this reason it is important to be able to learn both features if we hope to improve our chances at OOD generalization. Recent methods such as Saito et al. (2017), Saito et al. (2018), Zhang et al. (2021), Nam et al. (2020) and Liu et al. (2021a) all fail in this challenging scenario, we explain why in the following subsections E.1 to E.6. In E.7, we add a comparison between D-BAT and the concurrent work of Lee et al. (2022).

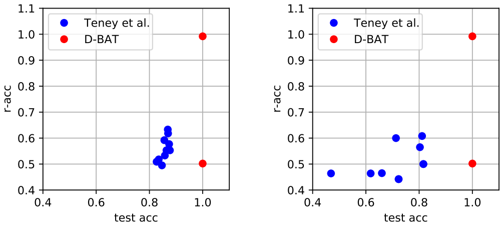
E.1 Comparison with Teney et al. (2021)

In their work, Teney et al. (2021) add a regularisation term $\boldsymbol{\delta}_{g_{\boldsymbol{\varphi}_1},g_{\boldsymbol{\varphi}_2}}$ which, given an input $\boldsymbol{x}$, is promoting orthogonality of hidden representations $\boldsymbol{h} = f_{\boldsymbol{\theta}}(x)$ given by an encoder $f_{\boldsymbol{\theta}}$ with parameters $\boldsymbol{\theta}$, and pairs of classifiers $g_{\boldsymbol{\varphi}_1}$ and $g_{\boldsymbol{\varphi}_2}$ of parameters $\boldsymbol{\varphi}_1$ and $\boldsymbol{\varphi}_2$ respectively:

$$\boldsymbol{\delta}_{g_{\boldsymbol{\varphi}_1},g_{\boldsymbol{\varphi}_2}} = \nabla_{\boldsymbol{h}} g^{\star}_{\boldsymbol{\varphi}_1}(x) \cdot \nabla_{\boldsymbol{h}} g^{\star}_{\boldsymbol{\varphi}_2}(x) \tag{T}$$

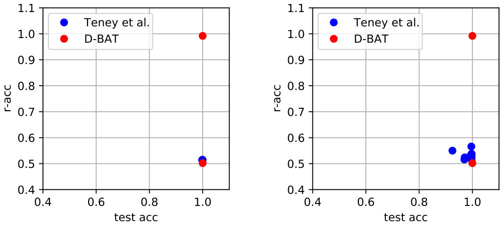
With $\nabla g^{\star}$ the gradient of its top predicted score.

We implemented the objective of Teney et al. (2021) with two different encoders: $f_{\boldsymbol{\theta}}(x) = x$ (identity) and a two-layers CNN. We tested it on our MM-Dominoes dataset (§ 4.1). The classification heads are trained simultaneously. Considering two classifications heads, we find two sets of hyperparameters, one that is giving the best compromise between accuracy and randomized-accuracy, and one that is keeping the accuracy close to 1. In the first setup in Fig. 6, we observe that none of the pairs of models trained with (T) as regulariser are particularly good at capturing any of the two features in the data. In contrast with D-BAT (with $\mathcal{D}_{\text{ood}}^{(1)}$) which is able to learn a second model having both high accuracy and high randomized-accuracy, hence capturing with the first model the two data modalities. For the second set of hyperparameters in Fig. 7, we observe that the improvement in randomized accuracy is only marginal if we do not want to sacrifice accuracy. We believe those results are explained by the many ways gradients of a neural network can be orthogonal while still encoding identical information. Better results might require training more classification heads (up to 96 heads are used in Teney et al. (2021).



(a) Identity encoder (b) CNN encoder

Figure 6: Comparison between D-BAT and Teney et al. (2021) with hyperparameters favoring the compromise between accuracy (test-acc) and randomized-accuracy (r-acc). We run 5 different seeds for Teney et al. (2021), each run consisting in two classification heads and a shared encoder chosen to be the identity (a) or a CNN encoder (b). The acc and r-acc are displayed for the 10 resulting classification heads. We compared with two models obtained using D-BAT, the first model learning the simplest feature is in the bottom right corner, and the second model trained with diversity is in the top right corner. We observe that the method of Teney et al. (2021) is failing to reach a good r-acc, and is sacrificing accuracy. D-BAT is able to retrieve both data modalities without sacrificing accuracy.



(a) Identity encoder (b) CNN encoder

Figure 7: Comparison between D-BAT and Teney et al. (2021) with hyperparameters yielding an accuracy (test-acc) close to 1 while maximizing the randomized-accuracy (r-acc). We run 5 different seeds for Teney et al. (2021), each run consisting in two classification heads and a shared encoder chosen to be the identity (a) or a CNN encoder (b). The acc and r-acc are displayed for the 10 resulting classification heads. We compared with two models obtained using D-BAT, the first model learning the simplest feature is in the bottom right corner, and the second model trained with diversity is in the top right corner. We observe that the method of Teney et al. (2021) is only marginally improving the randomized-acc.

E.2 Comparison with Zhang et al. (2021)

In their work, Zhang et al. (2021) argue that while a model can be biased, there exist unbiased functional subnetworks. They introduced Modular Risk Minimization (MRM) to find those subnetworks. We implemented the MRM method (Alg.1 from their paper) and tested it on our MM-Dominoes dataset (§ 4.1). We observed that their approach cannot handle the extreme case we consider where the spurious feature is fully predictive in the train distribution (but not in OOD). They need it to be, say, only 90% predictive. On our dataset, in the first phase of Alg.1, the model trained on the source task learns to completely ignore the bottom row due to the extreme simplicity bias, ensuring there is no useful sub-network. We found the randomized-accuracy of subnetworks obtained with MRM to be no better than random. This is because, in extreme cases, the network which the simplicity bias pushes us to learn may completely ignore the actual feature and instead only focuses on the spurious feature. In such a case, there is no un-biased subnetwork.

E.3 Comparison with Saito et al. (2017)

Contrary to Saito et al. (2017), we aim to train an ensemble of predictors able to generalize to *unknown* target tasks and do not assume access to the target data. In particular, the unlabelled OOD data we need can be different from the downstream transfer target data. We make this distinction clear in § 4.1 where $\mathcal{D}_{\text{ood}}^{(3)}$ for the dominoes datasets are built using combinations of 1s and 0s with images from classes *not present* in the target and source tasks. Despite the lack of target data, the r-acc improves by resp. 28% and 38% for the MM-Dominoes and MF-Dominoes datasets. Further, we focus on mitigating extreme *simplicity bias* as described by Shah et al. (2020), where a spurious feature can have the same predictive power as a non-spurious one on the source task (but not on the unknown target task). While (Saito et al., 2017) uses the concept of diversity, their formulation measures diversity in temrs of the inner-product between the weights. However, since neural networks are highly non-convex, it is possible for two networks to effectively learn the exact same function which relies on spurious features, while still having different parameterization. Thus, our method can be viewed as "functional" extension of the method in (Shah et al., 2020). Further, the encoder F itself can learn a representation such that F_1 and F_2 rely on the same information while minimizing the regularizer.

To see this, we trained the method of Alg.1 from (Saito et al., 2017) on our MM-Dominoes dataset. Tuning $\lambda \in \{0.1, 1, 10, 100\}$, we were unable to learn a model F_t which transfers to the target task.

E.4 Comparison with Saito et al. (2018)

Contrary to Saito et al. (2018), we do not aim at training a domain agnostic representation, but instead on overcoming simplicity bias to generalize to OOD settings. E.g. in colored MNIST, a classifier which throws out the shape and simply uses color (or vice-versa) is domain agnostic. But for overcoming spurious features, models in our ensemble would need to use *both* color and digit. Thus a domain agnostic representation is insufficient for OOD generalization.

Furthermore, the training procedure of (Saito et al., 2018) consists in first training a shared feature extractor G and two classification heads F_1 and F_2 to minimize the cross-entropy on the source task. In a second step the classification heads F_1 and F_2 are trained to increase the discrepancy on samples from the target distribution while fixing the feature extractor G. However, in the case where a spurious feature is as predictive as the non-spurious one — as in our experiments of § 4.1 — the extreme simplicity bias would force the feature extractor to become invariant to the complex feature. The second and third steps of the algorithm would fail from there.

E.5 Comparison with Nam et al. (2020)

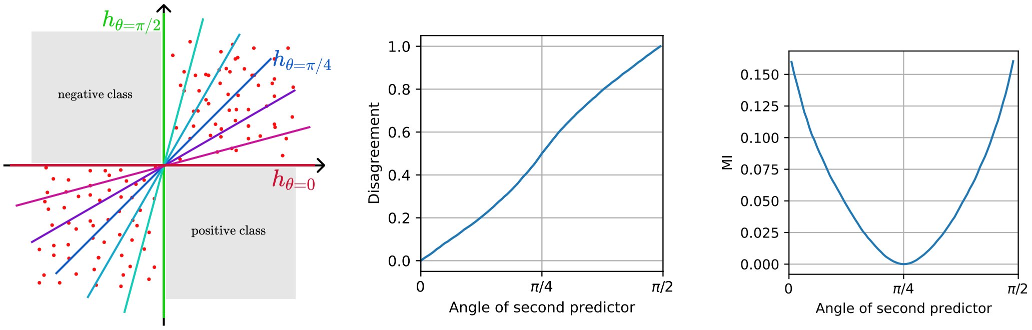
In this work, two models are trained simultaneously, one being the biased model while the other is the debiased model. During training, the first model gives higher weights to training samples agreeing with the current bias of the model. On the other hand, the second model learns by giving higher weights to training samples conflicting with the biased model. In order to work, the algorithm considers that the ratio of bias-aligned samples is smaller than 100%, which is not the case for our datasets in § 4.1). In these challenging datasets, where the biased feature is as predictive as the not biased feature, the second model fails to find bias-conflicting samples, hence would fail to de-biased itself. For this reason, the work of Nam et al. (2020) fails to counter extreme simplicity bias.

E.6 Comparison with Liu et al. (2021a)

The work of Liu et al. (2021a) is similar to the work of Nam et al. (2020) and shares the same limitation. A first model is trained through ERM before a second model trained by upweighting the samples misclassified in by the first model. This method, as for Nam et al. (2020), is failing to induce diversity when all the samples are correctly classify by the first model, as this is the case for our datasets in § 4.1.

E.7 Comparison with Lee et al. (2022)

The concurrent work of Lee et al. (2022) proposes to measure diversity between two models using the mutual information (MI) between their predictions on the entire OOD distribution, whereas our loss is defined on the per datapoint difference in the predictions. This means that our loss decomposes as a sum over the data-points and is well defined on small mini-batches. Computing the mutual information (MI) needs processing the entirety(or at least a very large part) of the data. Besides such practical advantages, our notion of diversity naturally arises out of discrepancy based domain adaptation theory, whereas the choice of using MI is ad-hoc and in fact may not give the expected results. Consider the toy-problem in Fig.3 of Lee et al. (2022) - the predictions of the two models actually have maximum mutual information since they predict the exact opposite on all the unlabelled perturbation data. Thus, MI would say that the two models actually have zero diversity, whereas discrepancy would say they have very high diversity. Hence, MI is theoretically the wrong measure to use. We confirmed this intuition by running experiments on the same setup as in Lee et al. (2022), we compared for the two notions of diversity (MI and discrepancy) which pairs of predictor are optimal. Results can be seen in Fig. 8.



(a) Experimental setup

(b) Disagreement

(c) Mutual Information

Figure 8: Disagreement and mutual information of potential second models h_2. In **(a)** we summarize the experimental setup which is similar to Fig.3 of Lee et al. (2022). The training data consists of the diagonal regions of $[0, 1] \times [-1, 0]$ as class 1 (positive), and $[-1, 0] \times [0, 1]$ as class 2 (negative). OOD datapoints $\tilde{X}$ are sampled randomly in the off-diagonal $[-1, 0]^2$ and $[0, 1]^2$ regions. The set of hyperplanes h_θ with $\theta \in [0, \pi/2]$ all achieve a perfect train accuracy. We fix the first classifier to be the horizontal $h_1 = h_{\theta=0}$ classifier. Then, we measure the disagreement between h_1 and different choices of $h_2 = h_\theta$ (in **b**), as well as their mutual information (in **c**) using the code provided in (Lee et al., 2022). Maximizing the disagreement yields the correct vertical classifier $h_2 = h_{\theta=\frac{\pi}{2}}$, whereas minimizing mutual information would yield the *wrong diagonal classifier*. The disagreement scores match intuitive definitions of diversity, whereas mutual information does not.